# Dynamic Programming Algorithm for Optimization of $\beta$-Decision Rules

Talha Amin[1], Igor Chikalov[1], Mikhail Moshkov[1], and Beata Zielosko[1,2]

[1] Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{talha.amin,igor.chikalov,mikhail.moshkov,beata.zielosko}@kaust.edu.sa
[2] Institute of Computer Science, University of Silesia
39, Będzińska St., 41-200 Sosnowiec, Poland

**Abstract.** We describe a dynamic programming algorithm that builds a decision rule for each row of a decision table. Rules are constructed according to the minimum description length principle: they have minimum length. Model complexity is regulated by adjusting rule accuracy.

**Key words:** decision rules, dynamic programming, minimization of decision rule length

## 1 Introduction

Decision rules are widely used for representation of knowledge extracted from data sets and for construction of classifiers that predict characteristics of new objects on the basis of information on existing objects [13, 19].

Exact decision rules can be "overlearned", i.e., may depend on the "noise" present in the input data. Therefore, recent years particular attention has been devoted to approximate decision rules (which can be considered also as approximate local reducts) and which are studied intensively by H.S. Nguyen, A. Skowron, D. Ślęzak, Z. Pawlak, J. Wróblewski and others [2, 9–12, 14, 15, 21–23].

In this paper, we consider one more approach to the definition of notion of approximate decision rule: we study so-called $\beta$-decision rules.

Models with shorter description are commonly believed to be more appropriate among the models with similar accuracy (minimum description length principle [18]). Following this principle, we are interested in building the shortest $\beta$-decision rules.

In this paper, we study dynamic programming approach to optimization of $\beta$-decision rules. Similar approach (but for optimization of another kind of approximate rules) was considered in [4, 25]

There are different approaches to construction of decision rules: brute-force approach which is applicable to tables with relatively small number of attributes, genetic algorithms [22, 24], simulated annealing [6], Boolean reasoning [11, 16, 20], ant colony optimization [3, 7], algorithms based on decision tree construction

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 10-16

[8, 10, 17], different kinds of greedy algorithms [9, 11]. Each method can have different modifications. For example, as in the case of decision trees, we can use greedy algorithms based on Gini index, entropy, etc., for construction of decision rules.

The most part of mentioned approaches (with the exception of brute-force one and Boolean reasoning) cannot guarantee the construction of shortest rules. We prove that our algorithm based on dynamic programming constructs $\beta$-decision rules with minimum length.

The paper consists of five sections. In the second section, we discuss main notions. In the third section, we study dynamic programming algorithm for minimization of $\beta$-decision rule length. In the fourth section, we discuss results of experiments for datasets from [5]. The fifth section contains short conclusion.

## 2 Decision Tables and $\beta$-Decision Rules

Decision table is a rectangular table $T$ with $n$ columns filled with nonnegative integers. Columns of the table are assigned attributes $f_1, \ldots, f_n$. Table rows are pairwise different, and each row $r$ is labeled with a decision – a nonnegative integer. Table rows are interpreted as tuples of attribute values.

A decision attached to the maximum number of rows of $T$ is called the most common decision for $T$. If we have two or more such decisions, we choose the smallest one. Denote by $R(T)$ the number of unordered pairs of rows from $T$ with different decisions.

A subtable of $T$ is a table obtained from $T$ by removing some rows with their assigned decisions. Let $j(1), \ldots, j(k) \in \{1, \ldots, n\}$ and $b_1, \ldots, b_k$ be nonnegative integers. Denote by $T(f_{j(1)}, b_1) \ldots (f_{j(k)}, b_k)$ the subtable of the table $T$, containing the rows from $T$, which at the intersection with the columns $f_{j(1)}, \ldots, f_{j(k)}$ have numbers $b_1, \ldots, b_k$ respectively. These subtables, including the table $T$, are called separable subtables of $T$.

Let $r = (a_1, \ldots, a_n)$ be a row of the table $T$, $i(1), \ldots, i(m) \in \{1, \ldots, n\}$, and $\beta$ be a real number such that $0 \leq \beta < 1$. The expression

$$f_{i(1)} = a_{i(1)} \wedge \ldots \wedge f_{i(m)} = a_{i(m)} \rightarrow d \tag{1}$$

is called a $\beta$-decision rule for $r$ and $T$, if $d$ is the most common decision for $T' = T(f_{i(1)}, a_{i(1)}) \ldots (f_{i(m)}, a_{i(m)})$ and $R(T') \leq \beta R(T)$. The number $m$ is called the length of this decision rule.

Let $\Theta$ be a subtable of the table $T$, and $r$ a row of $\Theta$. We say that (1) is a $(\beta, T)$-decision rule for $r$ and $\Theta$, if $R(\Theta(f_{i(1)}, a_{i(1)}) \ldots (f_{i(m)}, a_{i(m)})) \leq \beta R(T)$. A decision rule

$$\rightarrow d \tag{2}$$

of length 0 is a $(\beta, T)$-decision rule for $r$ and $\Theta$ if and only if $R(\Theta) \leq \beta R(T)$ and $d$ is the most common decision for $\Theta$. Note that the notion of $(\beta, T)$-decision rule for $r$ and $T$ coincides with the notion of $\beta$-decision rule for $r$ and $T$.

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 10-16

# 3 Dynamic Programming Algorithm for Optimization of Decision Rules

## 3.1 Construction of Graph $G_\beta(T)$

In this subsection, we describe an algorithm for constructing a directed acyclic graph $G_\beta(T)$ that is used in the rule optimization procedure.

Let $\Theta$ be a subtable of the table $T$. Denote by $E(\Theta)$ the set of attributes $f_i$, for which at least two numbers are different in the column $f_i$ of the table $\Theta$. For $f_i \in E(\Theta)$, denote by $E(\Theta, f_i)$ the set of numbers contained in the column $f_i$ of the table $\Theta$.

Consider an algorithm that constructs a directed acyclic graph $G_\beta(T)$. Nodes of the graph are separable subtables of the table $T$. During each step, the algorithm processes one node and marks it with the symbol *. At the first step, the algorithm constructs a graph containing a single node $T$.

Let the algorithm have already performed $p$ steps. Let us describe the step $(p+1)$. If all nodes are marked with the symbol * as processed, the algorithm finishes its work and presents the resulting graph as $G_\beta(T)$. Otherwise, choose a node (table) $\Theta$, which has not been processed yet. If $R(\Theta) \leq \beta R(T)$, then mark $\Theta$ with the symbol * and go to the step $(p+2)$. Let $R(\Theta) > \beta R(T)$. For each $f_i \in E(\Theta)$, draw a bundle of edges from the node $\Theta$. Let $E(\Theta, f_i) = \{b_1, \ldots, b_t\}$. Then draw $t$ edges from $\Theta$ and label these edges with pairs $(f_i, b_1), \ldots, (f_i, b_t)$ respectively. These edges enter to nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$. If some of nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$ are absent in the graph then add these nodes to the graph. Mark the node $\Theta$ with the symbol * and proceed to the step $(p+2)$.

Note that $E(\Theta) = \emptyset$ implies $\Theta$ contains a single row (due to the fact that the rows of $T$ are pairwise different) and, therefore, $R(\Theta) = 0 \leq \beta R(T)$.

## 3.2 Procedure of Optimization

Let us describe a procedure of optimization, that for each node $\Theta$ in the graph $G_\beta(T)$ assigns to each row $r$ of $\Theta$ a $(\beta, T)$-decision rule for $r$ and $\Theta$.

We will move from the terminal nodes of the graph $G_\beta(T)$ which do not have any outgoing edges ($R(\Theta) \leq \beta R(T)$ for each node $\Theta$ of the kind), to the node $T$.

Let $\Theta$ be a terminal node. Assign to each row $r$ of the table $\Theta$ the decision rule (2) of length 0, where $d$ is the most common decision for $\Theta$.

Let $\Theta$ be a nonterminal node and all children of $\Theta$ already have decision rules assigned to their rows. Let us assign a decision rule to each row $r$ of the table $\Theta$. Let $r = (a_1, \ldots, a_n)$. Consider all children of $\Theta$ containing the row $r$ and choose among them a subtable $\Theta(f_i, a_i)$ that has a shortest decision rule $\alpha \to d'$ assigned to $r$. Assign the rule $\alpha \wedge f_i = a_i \to d'$ to the row $r$ in the table $\Theta$.

**Theorem 1.** *For any node $\Theta$ of the graph $G_\beta(T)$ and any row $r$ of the table $\Theta$, the decision rule assigned to $r$ by the optimization procedure is a $(\beta, T)$-decision rule for $r$ and $\Theta$ having minimum length.*

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 10-16

*Proof.* We will prove this statement by induction on nodes (subtables) from $G_\beta(T)$. It is clear that for each terminal node the considered statement is true.

Let $\Theta$ be a nonterminal node and for all children of $\Theta$ the considered statement hold. Let $r = (a_1, \ldots, a_n)$ be a row of $\Theta$. Then for some $f_i \in E(\Theta)$ the row $r$ is labeled with a decision rule $f_i = a_i \wedge \alpha_i \to d$ where $\alpha_i \to d$ is the decision rule attached to row $r$ in the table $\Theta(f_i, a_i)$. According to the inductive hypothesis, the rule $\alpha_i \to d$ is a $(\beta, T)$-decision rule for $r$ and $\Theta(f_i, a_i)$. Therefore the rule $f_i = a_i \wedge \alpha_i \to d$ is a $(\beta, T)$-decision rule for $r$ and $\Theta$.

Let us assume that there exists a shorter decision rule than $f_i = a_i \wedge \alpha_i \to d$ which is a $(\beta, T)$-decision rule for $r$ and $\Theta$. Since $\Theta$ is a nonterminal node (and therefore $R(\Theta) > \beta R(T)$), the left-hand side of this shorter rule should contain an equality of the kind $f_j = a_j$ for some $f_j \in E(\Theta)$. Thus this rule can be represented in the form $f_j = a_j \wedge \alpha \to d'$. Since this rule is a $(\beta, T)$-decision rule for $r$ and $\Theta$, the rule $\alpha \to d'$ is a $(\beta, T)$-decision rule for $r$ and $\Theta(f_j, a_j)$. According to the inductive hypothesis, the row $r$ in the table $\Theta(f_j, a_j)$ is labeled with a rule $\gamma \to d''$ which length is at most the length of the rule $\alpha \to d'$. From here and from the description of the optimization procedure it follows that the row $r$ in $\Theta$ is labeled with a rule which length is at most the length of the rule $f_j = a_j \wedge \gamma \to d''$ which is impossible. Therefore the rule attached to the row $r$ in $\Theta$ has the minimum length among all rules which are $(\beta, T)$-decision rules for $r$ and $\Theta$ . □

**Corollary 1.** *After completing the optimization procedure each row $r$ of the table $T$ in the graph $G_\beta(T)$ is assigned with a $\beta$-decision rule for $r$ and $T$ having minimum length.*

## 4 Preliminary Experimental Results

First, we consider some experimental results described in [4, 25]. The purpose of the experiments was to construct graphs $G_0(T)$ for some decision tables $T$ from [5] and to count the number of nonterminal nodes in these graphs (see Table 1). The obtained results show that for some relatively small decision tables the

**Table 1.** Number of nonterminal nodes in graph $G_0(T)$

| Decision table | Number of columns | Number of rows | Number of nonterminal nodes |
|---|---|---|---|
| MUSHROOM | 22 | 8124 | 58800 |
| POKER-HAND-TRAINING-TRUE | 10 | 25010 | 1426236 |
| SPECT_ALL | 22 | 267 | 1089352 |
| CARS | 6 | 1728 | 3008 |
| NURSERY | 8 | 12960 | 53716 |
| TIC-TAC-TOE | 9 | 958 | 26678 |

graph $G_\beta(T)$ can be built even for $\beta = 0$. For larger tables the considered method is applicable only for values of $\beta$ which are closer to 1.

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 10-16

**Table 2.** Characteristics of DAG $G_\beta(T)$ for the decision tables MUSHROOM and CARS

| $\beta$ | MUSHROOM | | CARS | |
|---|---|---|---|---|
| | # nodes | # edges | # nodes | # edges |
| 0 | 149979 | 2145617 | 7007 | 19886 |
| 0.001 | 59802 | 619898 | 1727 | 3420 |
| 0.01 | 24648 | 161983 | 473 | 688 |
| 0.1 | 4148 | 750874 | 118 | 129 |
| 0.2 | 1692 | 5302 | 22 | 21 |

We now consider results of experiments described in [1] with two datasets from [5]: MUSHROOM (8124 rows and 22 conditional attributes) and CARS (1728 rows and 6 conditional attributes). For $\beta \in \{0, 0.001, 0.01, 0.1, 0.2\}$ we found the number of nodes in the directed acyclic graph $G_\beta(T)$ and the number of edges in $G_\beta(T)$ (see Table 2). The obtained results show that the number of nodes and the number of edges for $G_\beta(T)$ decrease with the growth of $\beta$. It means that the parameter $\beta$ can be used for managing algorithm complexity. They show also that the structure of graph $G_\beta(T)$ is usually far from a tree: the number of edges is essentially larger than the number of nodes.

Tables 3 and 4 contain results of experiments with three decision tables from [5]: BALANCE SCALE, SHUTTLE LANDING, and TIC-TAC-TOE. For each row of

**Table 3.** Length of optimal rules

| Dataset | parameters | | $\beta = 0$ | | | $\beta = 0.01$ | | | $\beta = 0.05$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | # attr. | # rows | min | avg | max | min | avg | max | min | avg | max |
| BALANCE SCALE | 4 | 625 | 3 | 3.197 | 4 | 2 | 2.000 | 2 | 1 | 1.000 | 1 |
| SHUTTLE LANDING | 6 | 15 | 1 | 1.400 | 4 | 1 | 1.400 | 4 | 1 | 1.200 | 4 |
| TIC-TAC-TOE | 9 | 958 | 3 | 3.017 | 4 | 2 | 2.001 | 3 | 1 | 1.292 | 2 |

**Table 4.** Length of optimal rules

| Dataset | $\beta = 0.1$ | | | $\beta = 0.2$ | | | $\beta = 0.3$ | | | $\beta = 0.5$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | min | avg | max | min | avg | max | min | avg | max | min | avg | max |
| BALANCE SCALE | 1 | 1.000 | 1 | 1 | 1.000 | 1 | 1 | 1.000 | 1 | 1 | 1.000 | 1 |
| SHUTTLE LANDING | 1 | 1.133 | 3 | 1 | 1.067 | 2 | 1 | 1.000 | 1 | 1 | 1.000 | 1 |
| TIC-TAC-TOE | 1 | 1.081 | 2 | 1 | 1.000 | 1 | 1 | 1.000 | 1 | 1 | 1.000 | 1 |

these tables, for each value of $\beta \in \{0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$ we find a $\beta$-decision rule with minimum length (optimal rule). For each of the three decision tables, Table 3 and Table 4 contain minimum length, maximum length and average length of optimal decision rules. We can see that the average length

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 10-16

of exact decision rules for BALANCE SCALE and TIC-TAC-TOE is about 3 and is decreasing when $\beta$ is growing. Beginning from $\beta = 0.3$ minimum, average and maximum length of optimal rules for all datasets is equal to 1.

## 5 Conclusion

The paper is devoted to the consideration of one more type of approximate rules – $\beta$-decision rules. We designed an algorithm for minimization of $\beta$-decision rule length and considered results of preliminary experiments. If we study exact rules (0-decision rules), then this algorithm is applicable to some relatively small decision tables. For large tables, we can work only with $\beta$-decision rules, where $\beta$ should be not far from 1.

## References

1. Alkhalid, A., Chikalov, I., Hussain, S., Moshkov, M.: Extensions of dynamic programming as a new tool for decision tree optimization. In: Ramanna, S., Howlett, R.J., Jain, L.C. (eds.) Emerging Paradigms in Machine Learning, Springer (to appear)
2. Bazan, G.J., Nguyen, S.H., Nguyen, T.T., Skowron, A., Stepaniuk, J.: Decision rules synthesis for object classification. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis, pp. 23–57. Physica-Verlag, Heidelberg (1998)
3. Boryczka, U., Kozak, J.: New algorithms for generation decision trees – Ant-Miner and its modifications. Foundations of Computational Intelligence 6, 229–262 (2009)
4. Chikalov, I., Moshkov, M., Zielosko, B.: Online learning algorithm for ensemble of decision rules. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS (LNAI), vol. 6743, pp. 310–313. Springer, Heidelberg (2011)
5. Frank, A., Asuncion, A.: UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml`
6. Jensen, R., Shen, Q.: Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Transactions on Knowledge and Data Engineering 16, 1457–1471 (2000)
7. Liu, B., Abbass, H.A., McKay, B.: Classification rule discovery with ant colony optimization. In: IEEE/WIC International Conference on Intelligent Agent Technology, pp. 83–88. IEEE Computer Society, Washington, DC (2003)
8. Michalski, S.R., Pietrzykowski, J.: iAQ: A program that discovers rules. In: 22nd Conference on Artificial Intelligence. Vancouver, Canada (2007)
9. Moshkov, M., Piliszczuk, M., Zielosko, B.: Partial Covers, Reducts and Decision Rules in Rough Sets: Theory and Applications. Studies in Computational Intelligence, vol. 145. Springer, Heidelberg (2008)
10. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning: A Rough Set Approach. Studies in Computational Intelligence, vol. 360. Springer, Heidelberg (2011)
11. Nguyen, H.S.: Approximate Boolean reasoning: foundations and applications in data mining. In: Peters, J.F., Skowron, A. (eds.) LNCS Transactions on Rough Sets V. LNCS, vol. 4100, pp. 344–523. Springer, Heidelberg (2006)

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 10-16

12. Nguyen, H.S., Ślęzak, D.: Approximate reducts and association rules – correspondence and complexity results. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 137–145. Springer, Heidelberg (1999)
13. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
14. Pawlak, Z.: Rough set elements. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 10–30. Physica-Verlag, Heidelberg (1998)
15. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
16. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences 177, 41–73 (2007)
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
18. Rissanen, J.: Modelling by shortest data description. Automatica 14, 465–471 (1978)
19. Skowron, A.: Rough sets in KDD. In: 16th IFIP World Computer Congress, pp. 1–14. Publishing House of Electronic Industry, Beijing (2000)
20. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
21. Ślęzak, D.: Normalized decision functions and measures for inconsistent decision tables analysis. Fundamenta Informaticae 44, 291–319 (2000)
22. Ślęzak, D., Wróblewski, J.: Order-based genetic algorithms for the search of approximate entropy reducts. In: Wang, G., Liu Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 308–311. Springer, Heidelberg (2003)
23. Wróblewski, J.: Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae 47, 351–360 (2001)
24. Wróblewski, J.: Finding minimal reducts using genetic algorithm. In: 2nd Annual Join Conference on Information Sciences. Wrightsville Beach, NC, pp. 186–189 (1995)
25. Zielosko, B., Moshkov, M., Chikalov, I.: Optimization of decision rules based on methods of dynamic programming. Vestnik of Lobachevsky State University of Nizhni Novgorod 6, 195–200 (2010) (in Russian)

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 10-16